**Lab assignment -3(2403A52399):**

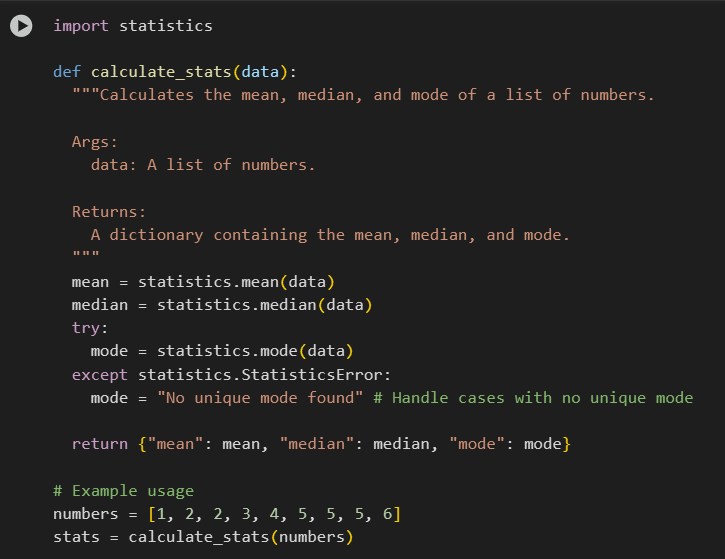
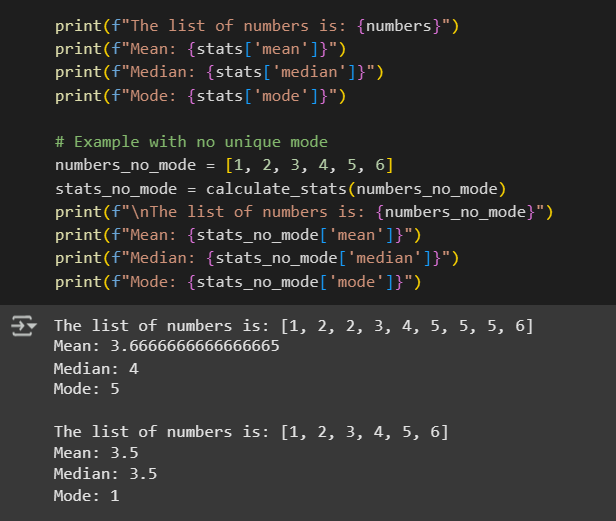
# PROMPT1:write a python program to calculate compount intrest by using function

# Output:

# Explanation: this code calculates compound interest. It asks you for the initial amount, interest rate, time, and how often the interest is added. It then uses a formula to figure out the final amount you'll have after that time and shows you the result

# PROMPT2:generate a python code to calculate the mean,median and mode of a list of numbers

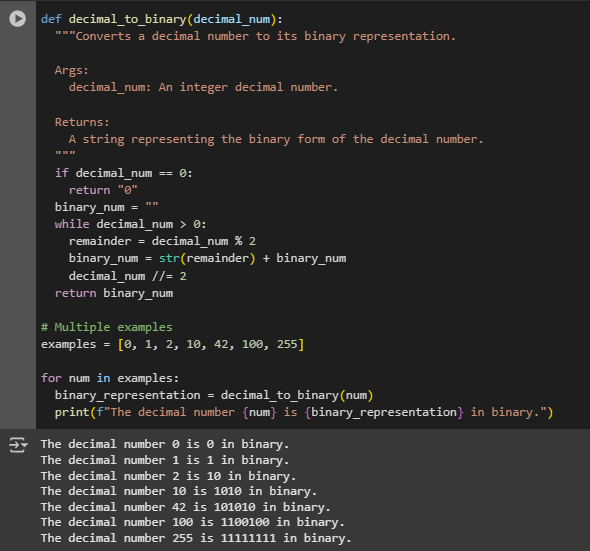
# OUTPUT:



# DESCRIPTION: This code uses Python's statistics module to find the mean, median, and mode of a list of numbers and prints the results. It also handles cases where there's no unique mode

# Prompt3: generate a python program to convery into binarynumbers with multiple examples

# Output:



# Description:This Python code defines a decimal\_to\_binary function . that takes a decimal integer and converts it into its binary equivalent as a string. The code then demonstrates this function by converting and printing the binary representations for a few example decimal numbers.

# Prompt4:Generate a python program to create a user interface for a hotel to generate bill based on the food items selected

# Code:

# 1. Define food items and prices (from step 1 of the current task)

menu = {

    "Burger": 8.99,

    "Pizza": 12.50,

    "Pasta": 10.00,

    "Salad": 7.50,

    "Fries": 3.00,

    "Soda": 2.50,

    "Water": 1.50,

    "Dessert": 5.00

}

# 2. Display the menu (from step 2 of the current task)

def display\_menu(menu):

  """Presents the food items and their prices to the user."""

  print("\nHotel Restaurant Menu:")

  for item, price in menu.items():

    print(f"- {item}: ${price:.2f}")

# 3. Get customer details (from step 3 of the current task)

def get\_customer\_details():

  """Prompts the user for customer details using console input."""

  print("\nPlease enter your details:")

  name = input("Name: ")

  contact = input("Contact Number: ")

  return {"name": name, "contact": contact}

# 4. Select food items (from step 4 of the current task)

def select\_food\_items(menu):

  """Allows the user to select food items from the menu and specify quantities.

  Args:

    menu: A dictionary where keys are food item names and values are their prices.

  Returns:

    A dictionary containing the selected items as keys and their quantities as values.

  """

  order = {}

  while True:

    item = input("Enter the name of the food item you want to order (or type 'done' to finish): ").strip().title()

    if item == 'Done':

      break

    if item in menu:

      while True:

        try:

          quantity\_str = input(f"Enter the quantity for {item}: ").strip()

          quantity = int(quantity\_str)

          if quantity > 0:

            break

          else:

            print("Quantity must be a positive integer.")

        except ValueError:

          print("Invalid input. Please enter a whole number for the quantity.")

      if item in order:

        order[item] += quantity

      else:

        order[item] = quantity

    else:

      print(f"'{item}' is not on the menu. Please select from the available items.")

    while True:

        add\_more = input("Do you want to add more items? (yes/no): ").strip().lower()

        if add\_more in ['yes', 'no']:

            break

        else:

            print("Invalid input. Please enter 'yes' or 'no'.")

    if add\_more == 'no':

        break

  return order

# 5. Calculate the bill (from step 5 of the current task)

def calculate\_bill(order, menu):

  """Calculates the total cost based on the selected food items and quantities.

  Args:

    order: A dictionary of selected items as keys and their quantities as values.

    menu: A dictionary of food item names as keys and their prices as values.

  Returns:

    The total calculated bill amount.

  """

  total\_bill = 0

  for item, quantity in order.items():

    if item in menu:

      total\_bill += menu[item] \* quantity

    else:

      # This case should ideally not be reached with proper input validation

      print(f"Warning: Item '{item}' not found on the menu.")

  return total\_bill

# 6. Display the bill (from step 6 of the current task)

def display\_bill(customer\_info, order, menu, total\_bill):

  """Presents the calculated bill to the user in a clear and organized format.

  Args:

    customer\_info: A dictionary containing customer details (name, contact).

    order: A dictionary of selected items as keys and their quantities as values.

    menu: A dictionary of food item names as keys and their prices as values.

    total\_bill: The total calculated bill amount.

  """

  print("\n" + "="\*30)

  print("      HOTEL BILL")

  print("="\*30)

  print("\nCustomer Details:")

  print(f"Name: {customer\_info.get('name', 'N/A')}")

  print(f"Contact: {customer\_info.get('contact', 'N/A')}")

  print("\nOrdered Items:")

  if order:

    for item, quantity in order.items():

      price\_per\_item = menu.get(item, 0)

      item\_cost = price\_per\_item \* quantity

      print(f"- {item}: {quantity} x ${price\_per\_item:.2f} = ${item\_cost:.2f}")

  else:

    print("No items ordered.")

  print("\n" + "-"\*30)

  print(f"Total Bill: ${total\_bill:.2f}")

  print("-" \* 30)

# Main program flow

if \_\_name\_\_ == "\_\_main\_\_":

  display\_menu(menu)

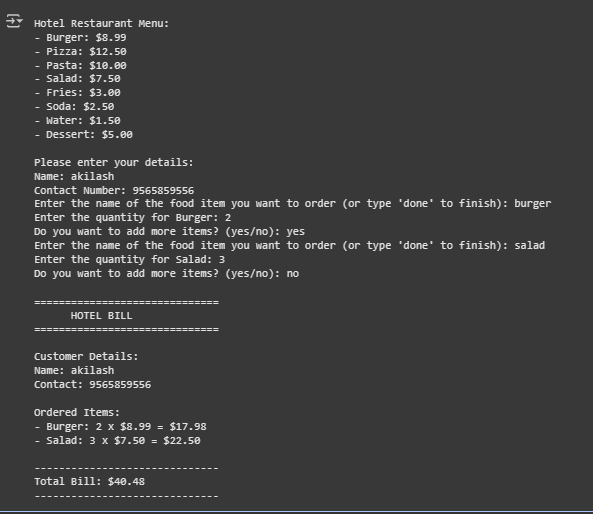
  customer\_info = get\_customer\_details()

  customer\_order = select\_food\_items(menu)

  total\_bill = calculate\_bill(customer\_order, menu)

  display\_bill(customer\_info, customer\_order, menu, total\_bill)

# output:



# Prompt5: generate a python program for Improving Temperature Conversion Function with Clear Instructions

# Code:

def convert\_temperature(value, unit):

  """Converts temperature between Celsius and Fahrenheit.

  Args:

    value: The temperature value to convert.

    unit: The current unit of the temperature ('C' for Celsius, 'F' for Fahrenheit).

  Returns:

    The converted temperature value and its new unit, or an error message if the unit is invalid.

  """

  if unit.upper() == 'C':

    # Convert Celsius to Fahrenheit

    fahrenheit = (value \* 9/5) + 32

    return fahrenheit, 'F'

  elif unit.upper() == 'F':

    # Convert Fahrenheit to Celsius

    celsius = (value - 32) \* 5/9

    return celsius, 'C'

  else:

    return "Invalid unit. Please use 'C' for Celsius or 'F' for Fahrenheit.", None

# --- Instructions ---

# 1. Call the convert\_temperature function with the temperature value and its current unit.

# 2. The unit should be 'C' for Celsius or 'F' for Fahrenheit.

# 3. The function will return a tuple containing the converted value and its new unit.

# 4. If an invalid unit is provided, it will return an error message and None for the unit.

# --- Examples ---

# Example 1: Convert 25 degrees Celsius to Fahrenheit

temp\_celsius = 25

converted\_temp\_f, unit\_f = convert\_temperature(temp\_celsius, 'C')

print(f"{temp\_celsius}°C is equal to {converted\_temp\_f:.2f}°{unit\_f}")

# Example 2: Convert 68 degrees Fahrenheit to Celsius

temp\_fahrenheit = 68

converted\_temp\_c, unit\_c = convert\_temperature(temp\_fahrenheit, 'F')

print(f"{temp\_fahrenheit}°F is equal to {converted\_temp\_c:.2f}°{unit\_c}")

# Example 3: Using an invalid unit

temp\_invalid = 100

converted\_temp\_invalid, unit\_invalid = convert\_temperature(temp\_invalid, 'K') # Using Kelvin as an example

print(converted\_temp\_invalid)

# Output:

